

## 1º Exercício: Torre de Controle

A palavra "cumbica", em tupi-guarani, significa "nuvem baixa" e, por coincidência, ☺ um dos aeroportos mais movimentados do país costuma ter nevoeiros que impedem decolagens e aterrissagens. Nessa situação, os voos destinados a Cumbica são desviados para outros lugares.

A partir de hoje, São José dos Campos começará a receber tais voos. Os aviões que chegam ficam "em órbita" e aguardam a torre de controle dar autorização de pouso. Para administrar esse intenso fluxo aéreo é necessário criar um *software* para controlar a **fila** dos aviões que aguardam permissão. Você foi contratado para essa tarefa!

Cada avião que chega à cidade envia um sinal à torre pedindo autorização para pouso. Os funcionários em terra avisam à torre cada vez que a pista fica liberada. Finalmente a torre, se a pista está livre, envia um sinal autorizando o pouso de alguma aeronave.

Você deve utilizar uma única **lista linear encadeada** para implementar a fila. No início da lista **não** use "nó cabeça". Em outras palavras: o primeiro elemento da fila deve ser apontado por um ponteiro simples. Se desejar, pode usar outro ponteiro (simples!) para o último elemento da fila.

As autorizações de pouso devem ser dadas na ordem de chegada (fila simples). Entretanto podem ocorrer situações de exceção. Por exemplo: uma aeronave em órbita pode ficar sem combustível. Nesse caso tal aeronave envia um novo sinal pedindo urgência. O programa da torre, mediante o novo pedido, deve reposicioná-la para o início da fila.

Importante: se mais de uma aeronave pedir urgência, a que fica no início da fila (próxima a pousar) deve ser a primeira que pediu urgência; a segunda da fila deve ser a segunda que pediu urgência e assim sucessivamente.

Antes de instalar o programa na torre, vamos fazer simulações com arquivos texto.

A entrada contém os sinais recebidos pela torre, a saber:

- pedidos de pouso de cada avião;
- informações de pista livre;
- pedidos de urgência para pouso.

Nenhum avião pede urgência se não tiver antes pedido pouso normal.

No arquivo de saída, o programa deve escrever, na ordem, os aviões que devem pousar.

Se porventura a pista ficar livre e não houver aviões em órbita, o programa deve informar "nenhum avião pousando". Após isso, os funcionários podem aproveitar para fazer manutenção na pista. Portanto, mesmo que chegue outra aeronave, a torre deve esperar o próximo aviso de pista livre.

Ao terminar o arquivo de entrada, o programa deve informar na saída a situação da fila.

**Entrada:**

`entrada1.txt`

Esse arquivo estará na mesma pasta que o programa.

Nas quatro primeiras linhas da entrada pode haver qualquer texto.

O restante do arquivo seguirá o formato do exemplo.

Cada linha inicial: máximo 70 caracteres.

Nome de cidade: máximo 30 caracteres.

**Saída:**

`Lab1_seu_nome_e_sobrenomes.txt`

Ou seja: o arquivo de saída deve ter o mesmo nome que o programa, mas extensão `.txt`

Nas quatro primeiras linhas, coloque qualquer texto, à sua escolha.

No restante do arquivo siga **exatamente** o formato do exemplo.

Exceção: Se no final a fila estiver vazia, escreva uma frase (livre ☺) informando isso.

**Importante:**

A cada pouso, o programa deve dar `free`.

**Observações:**

O programa **não** deve fazer pausa (`getchar, system ("pause")`).